# ANOTHER EFFICIENT ALGORITHM FOR CONVEX HULLS IN TWO DIMENSIONS

A.M. ANDREW
*Department of Cybernetics, University of Reading, Reading, England*

Received 30 April 1979; revised version received 10 September 1979

Convex hull

## 1. Introduction

Graham [8] devised a convex-hull algorithm depending on an initial ordering of the given set of points. Variations of his method giving increased efficiency have been described by Koplowitz and Jouppi [11] and by Anderson [2]. The purpose of the present note is to introduce another ordering algorithm which offers a further improvement in efficiency. First it is necessary to review some other developments in convex-hull algorithms to demonstrate that ordering algorithms are still worth considering.

The algorithm due to Jarvis [10] is usually represented as requiring a number of operations which is O(mn) where **n** is the number of points in the given set and **m** the number of sides of the convex hull. In fact this result is for a preliminary form of the algorithm which is then improved by the incorporation of a pointdeletion process. (Akl [1] shows that the point-deletion feature is actually essential.) Analysis of the efficiency of the improved algorithm is somewhat difficult. The upper-bound result only shows it to be preferable to an ordering algorithm when **m** is particularly small and experience indicates it is not more efficient in general.

The algorithm of Preparata and Hong [12], in its two-dimensional form, is not claimed to be more efficient than an ordering algorithm, though for large **n** it is only less efficient by a multiplicative constant.

The algorithm of Eddy [5,6] (see also [9]), to which that of Bykat [4] is similar, is claimed to represent an improvement on ordering algorithms. The basis of this claim requires careful consideration. An approximate analysis of the time requirement of his version of the algorithm is given by Bykat, based on an assumption that the algorithm is at its worst when all of the given points are vertices of a regular polygon. Fournier [7] shows this assumption to be false and describes another distribution of the points for which the algorithm would have a running time of $O(n^2)$. This worst-case distribution is special and improbable; Eddy considers operation of his version on some more realistic types of distribution.

The time taken by Eddy's algorithm is determined by the number of repetitions of an operation which tests whether a point is above or below a line. It is shown that this number is never greater than **mn**, the upper bound for the number of operations (of a different kind) required by the Jarvis algorithm.

Comparison of the Eddy algorithm with ordering algorithms depends on empirical data from trials on random sets of points ranging in size from 10 to 1000 and distributed in various ways in the plane.

Where the points were all on the boundary of a circle the algorithm performed rather poorly. For all of the other distributions used it was found that the number of repetitions of the operation was considerably smaller than n $\log_2 n$ and became a smaller fraction of it with increasing **n**. Let **k** be this fraction, so that the number of repetitions for some set of points is kn $\log_2 n$. Then values of **k**, averaged over the four types of distribution for which the algorithm is claimed to be superior, are calculated from Eddy's data as follows:

| n | 10 | 32 | 100 | 316 | 1000 |
|---|---|---|---|---|---|
| k | 0.632 | 0.602 | 0.492 | 0.345 | 0.337 |

In an ordering algorithm the ordering process alone requires n $\log_2 n$ operations. The time requirement of the other parts of the algorithm is O(n), so for sufficiently large **n** it becomes negligible in comparison. For large **n**, therefore, the comparison is between the time requirements of n $\log_2 n$ repetitions of the basic operation of a sorting process and kn $\log_2 n$ repetitions of the basic operation of Eddy's algorithm. The basic operation of Eddy's algorithm is much more time-consuming than that of a sorting process, since it requires a multiplication and an addition followed by a comparison, where the latter requires a comparison only. If, as is likely, the basic operation of Eddy's algorithm takes four times as long as the basic operation of a sorting process, his algorithm becomes more efficient than a sorting one only when **k** falls below 0.25.

The values for **k** tabulated above do not allow confident extrapolation to higher values of **n**. However, the closeness of the last two values suggests that **k** may converge to a limit greater than 0.3. It follows that the alleged superiority of Eddy's algorithm has not been established, and ordering algorithms are still worth considering.

A further important development is due to Bentley and Shamos [3]. They show that a worst-case time requirement of O(n log n) is inescapable, but they describe a method which achieves an *expected* time requirement of O(n). Their method must obviously be the method of choice for large **n**, but it does not eliminate interest in earlier methods, since it requires some other method to be applied repeatedly to subsets of the given set of points. Any improvement in the efficiency of the embedded algorithm must produce some improvement in the efficiency of the overall process.

## 2. Graham-type algorithms

The ordering algorithms have three parts:

(a) evaluation, for each of the **n** points, of a function of position to be used in ordering,

(b) the ordering process, and

(c) an iterative process to eliminate points which are not vertices of the convex hull.

The time requirement of (b) is O(n log n), while that of (a) and (c) is O(n). Since requirement (b) dominates for large **n** it is important that the comparisons to be made during ordering are of simple numbers and not, for example, of gradients combined with integers indicating quadrant. The time requirement for (b) is then the same for all the variations of Graham's method. The improvements introduced by Koplowitz and Jouppi [11] and by Anderson [2], as well as the improvement introduced here, affect only the O(n) requirements of (a) and (c).

Koplowitz and Jouppi suggest that the function computed in (a) should be a gradient, requiring a single operation of division for its evaluation. A function which is preferable since it combines the quadrant information and avoids the need to deal specially with infinite gradient is that computed, for points in the first two quadrants, as follows:

```
if x > y then y/x elif x > −y
   then 2 − x/y else 4 + y/x fi
```

Part (c) of the algorithm requires a number of convexity tests which is 2n in Graham's original algorithm and slightly less in the improved versions. Each test requires two multiplications. The requirements of (a) and (c) together, so far as the major arithmetic operations are concerned, can therefore be reduced to **n** divisions and approximately 4n multiplications.

## 3. The new algorithm

A saving can be achieved by eliminating part (a) of the algorithm altogether and ordering the points in terms of one of the Cartesian coordinates, say the y-coordinate. If there is more than one point having the minimum y-value, the leftmost and rightmost of them are identified, and similarly if there is more than one point having the maximum y-value. The convex hull has a left side joining the two leftmost points and a right side joining the two rightmost.

The points are formed into two chains, each of which is then subjected to an iterative point-elimination process so that one of them comes to represent the left side of the hull and the other the right side. The intermediate points are allocated to one or other of the chains according to the side on which they fall of a line joining the lowest point (or any one of the set of lowest points) to the highest point (or any one of the set of highest points).

If a point has x-coordinate smaller than that of the farther-left end of the dividing line it is added to the left-side chain, and conversely if its z-coordinate is greater than that of the farther-right end of the line. Only if the x-coordinate falls between those of the ends of the line, it is necessary to carry out a multiplication in testing for side of line on which the point falls. The number of multiplications to be carried out in allocating points to the two chains is at most n − 2 and usually much less.

Since the points are divided between the two chains the total number of multiplications required in the two point-elimination processes is similar to what is required in the single point-elimination process needed in the other ordering algorithms, namely a little less than 4n. The new method can therefore be said to eliminate the need for **n** divisions in part (a) of one of the earlier algorithms, at the cost of introducing an extra requirement for a number of multiplications which is not more than n − 2 and is usually much less. Since floating-point division in some machines takes significantly longer than floating-point multiplication there is a potential gain in efficiency.

## 4. Conclusion

It has been shown that ordering algorithms are still worth considering as efficient means of determining two-dimensional convex hulls, and a new efficient variation has been described.

A further saving in expected time could be achieved by an algorithm combining a part of Eddy's algorithm with a sorting algorithm. Such an algorithm might start by scanning the given set of points to locate the four having extreme values of each of the two coordinates. Then the points would be scanned again to eliminate those which were inside the quadrilateral having the four selected points as vertices. The time requirement of these two scans is O(n). Then an ordering algorithm could be applied to determine the hull of the reduced set of points, which would be the same as that of the full set. If the proportion of points remaining in the reduced set is not strongly dependent on **n**, the time requirement of the hybrid algorithm is still O(n log n) for large **n**, but it is more efficient than an ordering algorithm by a multiplicative constant.

The method can be extended by letting the number of extreme points selected be more than four, and eliminating points from the inside of the corresponding polygon. The optimal number of selected points must be a function of **n**, the total number of points.

Extension of the hybrid algorithm so that the number of selected points is automatically chosen as a function of **n** offers the possibility of an algorithm whose asymptotic expected time is other than O(n log n). Further study is necessary to determine what asymptotic behaviour can in fact be achieved and thus whether this approach can rival that of Bentley and Shamos [3].

## Acknowledgment

I am indebted to the referee of Information Processing Letters for bringing several recent references to my notice, and to my wife Joyce whose work on a statistical problem brought the convex-hull problem to my notice.

## References

[1] S.G. Akl, Two remarks on a convex hull algorithm, Information Processing Lett. 8 (1974) 108–109.

[2] K.R. Anderson, A reevaluation of an efficient algorithm for determining the convex hull of a finite planar set, Information Processing Lett. 7 (1978) 53–55.

[3] J.L. Bentley and M.I. Shamos, Divide and conquer for linear expected time, Information Processing Lett. 7 (1978) 87–91.

[4] A. Bykat, Convex hull of a finite set of points in two dimensions, Information Processing Lett. 7 (1978) 296–298.

[5] W.F. Eddy, A new convex hull algorithm for planar sets, ACM Trans. Math. Software 3 (1977) 398–403.

[6] W.F. Eddy, Algorithm 523: Convex, ACM Trans. Math. Software 3 (1977) 411–412.

[7] A. Fournier, Comments on convex hull of a finite set of points in two dimensions, Information Processing Lett. 8 (1979) 173.

[8] R.L. Graham, An efficient algorithm for determining the convex hull of a finite planar set, Information Processing Lett. 1 (1972) 132–133.

[9] P.J. Green and B.W. Silverman, Constructing the convex hull of a set of points in the plane, Comput. J. 22 (1979) 262–266.

[10] R.A. Jarvis, On the identification of the convex hull of a finite set of points in the plane, Information Processing Lett. 2 (1973) 18–21.

[11] J. Koplowitz and D. Jouppi, A more efficient convex hull algorithm, Information Processing Lett. 7 (1978) 56–57.

[12] F.P. Preparata and S.J. Hong, Convex hull of finite sets of points in two and three dimensions, Comm. ACM 20 (1977) 87–93.